- removing a processor of one of said processing sets; and
- replacing the removed processor with a replacement processor that includes:
    - an interface for communication with an I/O bus, and
    - a processor identification register which is read/writeable and has stored in said register data representative of a processor identification, wherein said replacement processor is responsive to a masking condition, to write a common predefined data value received via said I/O bus into said processor identification register, wherein said predefined data value is common to said processing sets and is operable to mask said processor identification.

REMARKS

Claims 18 were pending in the present application. Claims 1, 3, 6-12 and 15-16 have been amended. Accordingly, claims 1-18 remain pending in the application.

CONCLUSION

Applicants submit the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Conley, Rose, & Tayon, P.C. Deposit Account No. 501505/5181-80100/BNK.

Respectfully submitted,

B. Noël Kivlin
Reg. No. 33,929
ATTORNEY FOR APPLICANT(S)

Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 476-1400

Date: 6-24-02

Attachment 1, 5181-80100
Marked–up version of the amended claims.

1. (Amended)A computer system comprising

- a plurality of processing sets, each having at least one processor, and
- a bridge coupled to each of said processing sets and operable to monitor a step locked operation of said processing sets, wherein
- each of said processors has a processor identification register which is read/writeable and is operable to store in said register data representative of a processor identification, said processors being arranged, consequent upon a predetermined condition, to load a common predefined data value that is common to said processing sets into said processor identification register.

3. (Amended)A computer system as claimed in Claim 2, wherein said reset state is a state asserted in said [fault tolerant] computer system following boot or re-boot.

6. (Amended)A computer system as claimed in Claim 1, wherein said common predefined data value is an all zeros value.

7. (Amended)A computer system as claimed in Claim 1, wherein each of said processors further includes

- a read only register having stored therein said processor identification data.

8. (Amended)A computer system as claimed in Claim 7, wherein said processor identification data stored in said read only register is loaded, upon initialisation into said processor identification register.

9. (Amended)A computer system as claimed in Claim 1, wherein said common predefined value is a processor identification of one of the processors of said [fault tolerant] computer system, said processor identification of each of said processors being matched.

10. (Amended)A processor for use in a processing set forming part of a fault tolerant computer system that includes a plurality of processing sets, said processor comprising

- an interface for communication with an I/O bus, and
- a processor identification register which is read/writeable[,] and has stored in said register data representative of a processor identification, wherein said processor is [operable] responsive to a masking condition, to write a common predefined data value received via said I/O bus into said processor identification register, wherein said predefined data value is common to said processing sets and is operable to mask said processor identification.

11. (Amended)A processor as claimed in Claim 10, comprising

- a read only register having stored therein said processor identification data, wherein said processor identification data stored in said read only register is loadable, upon initialisation into said processor identification register.

12. (Amended)A method of operating a fault tolerant computer system comprising a plurality of processing sets, each of which processing sets is connected to a bridge, each of said processing sets having at least one processor, said method comprising the steps of;

a) detecting a predetermined condition representative of a state in which said processor identification is present in a processor identification register of said processor; and

b) loading a common predefined data value into said processor identification register of each of said processors, which predefined data value [has an effect of masking] is common to said processing sets and is operable to mask said processor identification.

15. (Amended)A method of operating a fault tolerant computer system as claimed in Claim 12, comprising

detecting an error condition of at least one of said plurality of processing sets, and

if said error condition is detected performing the step of loading said common predefined data value in said processor identification register of said at least one processor of the processing set which has said detected error.

16. A method of operating a fault tolerant computer system comprising a plurality of processing sets, each having at least one processor, and a bridge coupled to each of said processing sets and operable to monitor a step locked operation of said processing sets, said method comprising the steps of

- removing a processor of one of said processing sets; and
- replacing the removed processor with a replacement processor that includes:
    - an interface for communication with an I/O bus, and
    - a processor identification register which is read/writeable[,] and has stored in said register data representative of a processor identification, wherein said replacement processor is [operable] responsive to a masking condition, to write a common predefined data value received via said I/O bus into said processor identification register, wherein said predefined data value is common to said processing sets and is operable to mask said processor identification.

Attachment 2, 5181-80100
Marked-up version of Specification.

The paragraph beginning at Page 10, line 24 and ending at page 11, line 15:

"The presence of a different version identification number in a processor which forms part of a fault tolerant computing system described above provides a technical problem because processors within the fault tolerant computer system may have different version numbers although functionally and logically may operate substantially identically. An example of such a processor is the Sun [Microelectronics] Microsystems Ultra Sparc (Trademark) 1 and 2. As explained on page 241 of the Sun Ultra Sparc 1 and 2 Users Manual January 1997, the version register of the Ultra Sparc Processor includes two fields which identify the processor and identify the implementation of the processor as well as a mask set version. The fields of the version register are reproduced as an example of version identification data in the table shown in Figure 5. These fields will vary in accordance with a particular implementation of the processor. These fields form part of a 64 bit data word which include a field indicative of the maximum trap level supported (MAXTL) and a field indicative of the maximum number of windows of an integer register file (MAXWIN). As such, the version register will be interrogated by the operating system, and the contents may be stored in the memory units 56, 76 of the processing sets 14, 16. This is because the MAXTL and MAXWIN fields are required by the operating system in order to execute correctly. As a result there can exist in the memory units 56, 76 data values which will differ between the processing sets. As a result when the version identification data is loaded onto the I/O bus by the operating system, such as for example during a memory swap as part of a virtual memory access, a divergence will be caused between the data values which are put on to the PA-bus 24 and PB-bus 26. This may be detected by the bridge 12 as an error even though both processing sets 14, 16 of the computer system are operating correctly."